Microsoft

.net developer training

Create XML Web Services and Applications with Visual Studio .NET

Object Oriented Concepts in .NET

Module 06 Agenda

- Classes
- Object orientation
- Constructors
- Inheritance
- Namespaces and Assemblies

Classes Are Everywhere

- .NET Framework
- Your applications
  - Web Forms, Windows® Forms and Web services
  - Application components (Data layer and middle tier)
  - Utilities
- Can be hosted
  - In components
  - Directly in your application project
    - No separate DLL required

Classes

- **Classes contain members**
  - **Data members:**
    - **Variables and constants**
      - **For example:** `Public sCustomer as string`
    - **Properties**
      - **Values accessed through get/set methods**
  - **Methods:**
    - **Functionality for the object or class**
    - **Public GetCustomerByID(…) as …**

Paul Litwin:

should include vb on left, c# on right

tiating Classes

- Use new operator - Examples
  - Visual Basic

```
Dim obj  as New MyClass()
-or-
Dim obj  as MyClass
obj = New MyClass()
```

  - C#

```
MyClass obj = new MyClass();
-or
MyClass obj;
obj = new MyClass();
```

Classes Demo

- Simple Class
  - Create and instantiate a class
  - Create a simple component
  - Return a dataset from a component
  - Consume the component in the client

Object Orientation

- Supported by all .NET languages
  - Inheritance
  - Overloading
  - Interfaces
  - Shared members
  - Constructors and finalizers
  - Events and delegates

Paul Litwin:

again, C# would be nice to the right of vb way

ructors

- Constructor
  - Fired when object is created
  - An object's parameters can be overloaded (Method Parameter Overloading) For example:

```
anObject = New MyClass (ConnectionString)
```

Defined by:

Overloaded constructor

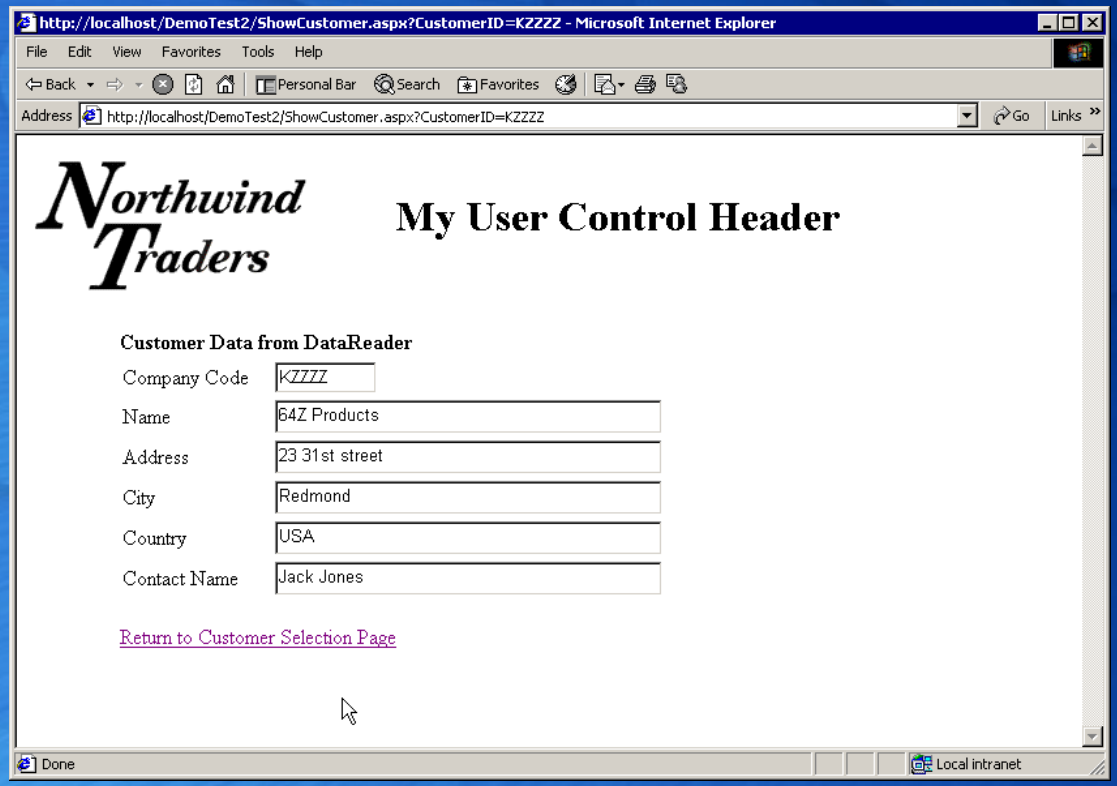
```
Public Sub New()
    'Requires connection string to be set elsewhere
End Sub

Public Sub New(ByVal sConnectString As String)

                gConnectString = sConnectString
```

Constructors Demo

- Data Class
- Show constructor and optional parameters and explain how they work



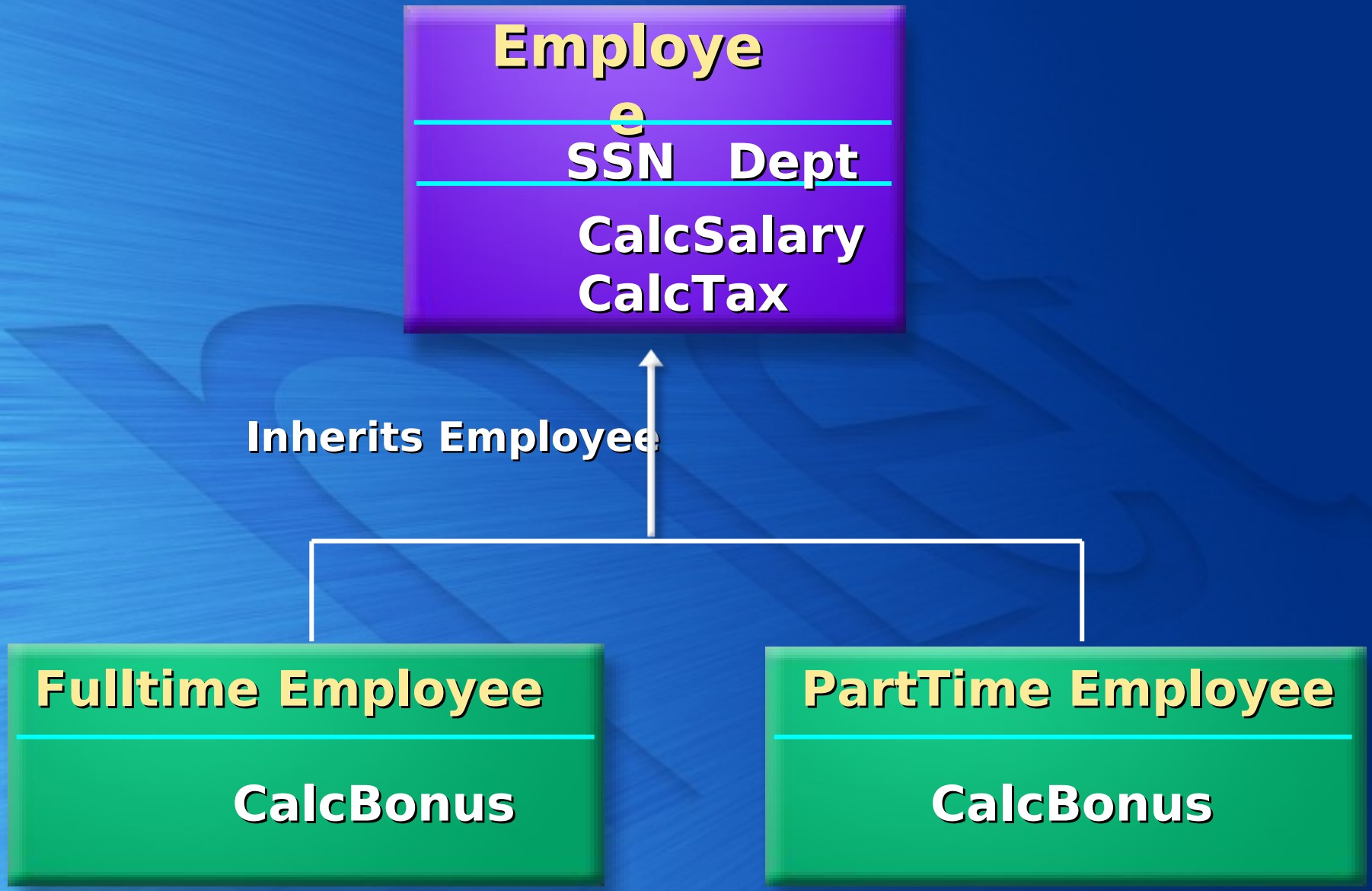
Paul Litwin:

Delete slide?

tance Concepts

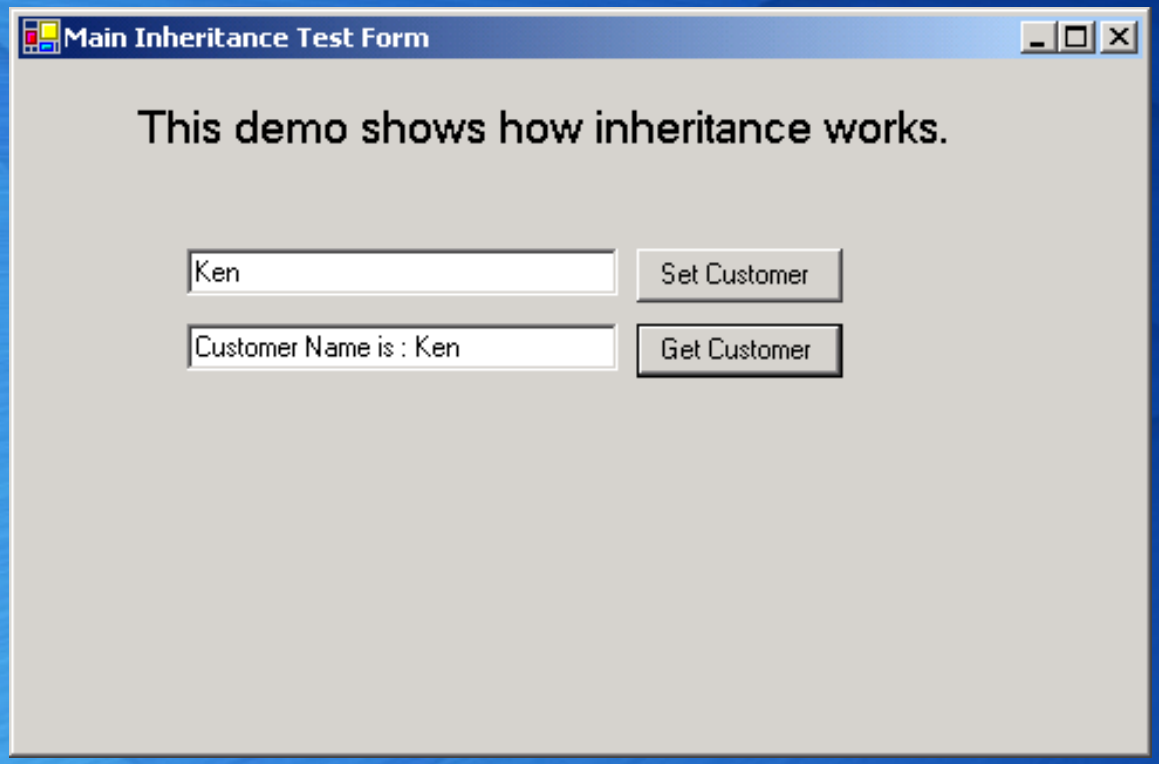
- Concept of reuse
- Building type hierarchies
- Versioning
- Polymorphism

How Inheritance Works



Inheritance Demo

- Simple Inheritance
- Demonstrate simple inheritance concepts



Paul Litwin:

I think it would be better to talk about Overrides and Overidable instead of this stuff (or in addition to)

Should we also talk about Overloads?

Again, anything we show should also be shown in C#

olling Inheritance

- ract base classes
  - creatable directly
  - st be inherited

```
ic MustInherit Class AbstractBase

Class
```

- inheritable classes
  - sses that should never be inherited

```
ic NotInheritable Class FinalClass
...
End Class
```

Namespaces and Assemblies

- Assemblies are the container for components
  - Similar to a COM DLL
  - Can be made of one or more files
  - Can be private or shared
- Namespaces provide a way to organize objects into hierarchies and minimize name collisions
  - Uses a dot-separated syntax (e.g., System.Data)
- Namespaces can be imported to reduce scoping

Paul Litwin:

Building what? a Component?

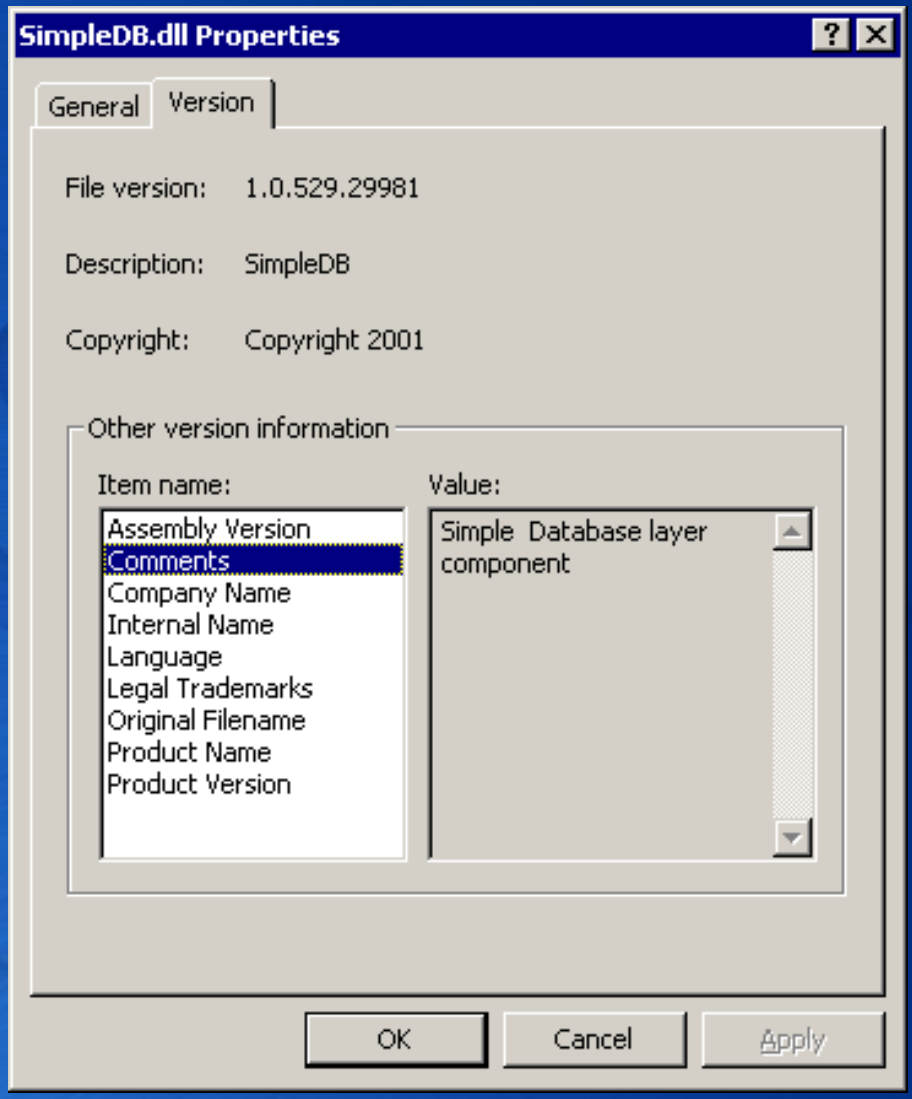
Better explanation of "Assembly level Custom..." needed

ng An Assembly

- bile code into assembly
  - embly is a unit of deployment for naged code
  - Manifest lists external dependencies
  - Versioning
  - Provide a strong name
- Assembly level custom attributes for:
  - COM+ application name
  - COM+ application description
  - COM+ application activation (Library/Server)
  - Assembly version
  - Assembly strong name

Assembly Demo

- Working with Assemblies
- Show how to add attributes to assembly.vb and build a component



Summary

- Business objects are easy to create in .NET
- The .NET base classes and inheritance are powerful tools to help you build your applications
- Assemblies and namespaces
  - Give you options on how to organize and reuse your components in assemblies
  - Allow for easy deployment
  - Strong names end DLL hell

Resources

- ***The GotDotNet Home Page*** **- An excellent site to find more information about .NET with samples and real-world examples of .NET applications. http://www.gotdotnet.com/**
- ***An introduction to the .NET Framework*** **- A nice, concise introduction to the .NET Framework. http://www.gotdotnet.com/about_new.aspx**

In-Depth Training On Components
Direct from Microsoft

- 2349A: Programming the Microsoft .NET Framework with C# (Prerelease)
- 2350A: Securing and Deploying Microsoft .NET Assemblies (Prerelease)
- 2415A: Programming the Microsoft .NET Framework with Visual Basic .NET (Prerelease)
- 2557A: Developing Component-Based Applications Using Microsoft .NET Enterprise Services
  - Available in May 2002

Microsoft®

06.21